IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of:

Benny Souder, et al.
(Appellants)

Serial No.: 10/718,747

Filed: November 21, 2003

For: AUTOMATIC AND DYNAMIC PROVISIONING OF DATABASES

Confirmation No. 1778

Examiner: Jean B. Fleurantin

Group Art Unit No.: 2162

Mail Stop Appeal Brief – Patents / via EFS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' APPEAL BRIEF

Sir:

Applicants/Appellants hereby submit this Appeal Brief in support of the Notice of Appeal filed on October 22, 2007.

I. REAL PARTY IN INTEREST

Oracle International Corporation, of Redwood Shores, California, is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1 – 9 and 18 – 31 and 40 – 44 are pending and are the subject of this appeal. Claims 10 – 17 and 32 – 39 are withdrawn from examination.

IV. STATUS OF AMENDMENTS

Amendments were filed after the Final Office Action. The amendments were not entered even though entry of the amendments is plainly proper. The amendments fixed dependency and antecedent problems, and can be addressed easily after appeal.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Claims 1 and 18 are independent. Independent claims 1 and 18 provide a similar solution for automatically provisioning or instantiating data in a distributed system. In claim 1, a database server causes a tablespace to be transported between file systems, in particular, a first file system and a second file system. (see present application, 0056 and FIG. 3, 0063 – 0065, FIG. 5) "A tablespace is a collection of storage containers (e.g. files) used to store data for database objects (e.g. relational tables)." (0013) After transporting the tablespace, the database server imports the tablespace into the database server's local database. (0056 and FIG. 3, 0063 – 0065, FIG. 5) Claim 18 is similar and supported in the specification in a similar way. A database server causes one or more files that store database data to be transported from a first file system to a second file system. The database server then provisions the database data as part of a database managed by the database server.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1 – 6, 18 – 28 and 40 – 44 are rejected under 35 USC 102(b) as being anticipated by "Bridge" (U.S. Patent No. 6,272,503).

2. Claims 7 – 9 and 29 – 31 are rejected under 35 USC 103(a) as being unpatentable over Bridge in view of Wang (U.S. Patent No. 5,758,345).

VII. ARGUMENT

A. Rejection of Claim 1 under 102(b)

Claim 1, recites:

a database server causing a tablespace to be transported from a first file system to a second file system; and

after transporting said tablespace to said second file system, said database server importing said tablespace into a local database managed by said database server.

To anticipate a claim, the reference must teach every element of the claim. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Id., Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628,631,2 USPQ2d 1051,1053

Claim 1 recites a database server that imports a tablespace into a local database managed by the database server. Importantly, the database server also causes the tablespace to be transported between a first and second file system. This particular way of automatically provisioning a database is not inherently much less expressly described by Bridge.

In support of the Examiner's position that Bridge describes the feature of a database server that causes a tablespace to be transported between file systems, the Examiner cites Bridge, at passage at col. 9, lines 41 – 50. In this passage, the Examiner must be correlating the target database to the claimed database server that imports the tablespace. Note the passage describes a source database system that produces a set of files in response to a user providing the name of the files.

Importantly, the user copies the files "to a place accessible to the target database [sic]." This teaching certainly fails to expressly describe a database server causing a tablespace to be

transported between file systems, as claimed. Further, the teaching also fails to inherently describe a database server causing a tablespace to be transported between file systems. It is not necessary for the database server to cause the tablespace to be transported because, for example, a user can manually invoke standard file transfer utilities, such as operating system commands or FTP, without involving the database server at all in transferring the files.

The Examiner further argues that Bridge teaches transferring a tablespace from a source database to a target database, and alleges that this teaching is read on by a database server causing the tablespace to be transported between file systems.

The Office Action is correct in that Bridge does teach transferring a tablespace from a source database to a target database. However, a general teaching about performing an operation does not by itself disclose or suggest each and every specific way of performing that operation. Disclosure of a genus does not necessarily disclose or suggest every species of the genus. It does not necessarily follow from the fact that a tablespace is transported between file systems that the transporting of tablespaces is being caused by a database server as claimed, for reasons given above.

The Office Action then recites the principle that the pending claims must be given the broadest reasonable interpretation consistent with the specification, citing In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). The principle is cited to justify a broad interpretation of the limitation "database server causes the tablespace to be transported between a first and second file system." Unfortunately, the Office Action never states what the interpretation is. Nevertheless, Applicant infers from the fact that the Office Action is relying upon the teaching in Bridge about transferring tablespaces between file systems to anticipate

claim 1, that the Office Action is interpreting claim 1 as reciting and claiming all ways of transporting tablespaces between file systems.

Such an interpretation ignores express limitations in claim 1, namely that the transporting of a tablespace between file systems is caused by a database server. Because the broad interpretation ignores an express limitation in a claim, the interpretation being applied by the Office Action is not reasonable.

Finally, the interpretation applied by an Examiner must also be consistent with the interpretation that those skilled in the art would reach. In re Cortight, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999) One skilled in the art would not interpret the limitation of a database server causing a tablespace to be transported between file systems to cover every way of transporting tablespaces or files. For example, one skilled in the art would not interpret a user invoking a file transfer utility of an operating system to copy files between file systems as being covered by the notion of a database server transporting tablespaces between file systems.

Based on the foregoing, Bridge fails to teach at least some features of claim 1, and therefore fails to teach all the features of claim 1. Therefore, claim 1 is patentable.

Rejection of Dependent Claim 3 Under 102(b)

Claim 3 was amended in the most previous amendment to fix an obvious antecedent basis error. The amendment was not entered. The amendment placed the claim in a better form for consideration on appeal and does not present new issues requiring further consideration and search. The amendment should have been entered.

Claim 3, as amended, recites "wherein said routine is written in code that conforms to a database language and that may be executed by a database server." A routine written in code that

conforms to a database language and that may be executed by a database server is not disclosed or suggested in any way by the cited art.

The Office Action alleges that this feature is described by Bridge teaching "executing by processor instructions." However, this teaching does not expressly or inherently require instructions that conform to a database language, such as PL/SQL.

Based on the foregoing, Bridge fails to teach at least some features of claim 3, and therefore fails to teach all the features of claim 3. Therefore, claim 3 is patentable.

Rejection of Independent Claim 18 Under 102(b)

Claim 18, recites a "method for automatically instantiating database data in a distributed database system", comprising

"a database server causing a set of one or more files to be transported from a first file system to a second file system;

wherein said set of one or more files store said database data; and

after transporting said set of one or more files to said second file system, said database server provisioning said database data as at least part of a database managed by said database server."

Claim 18 requires that to automatically instantiate a database, that a database server cause a file that stores data for a database to be transported between a first and second file system, and then provision the data as at least part of a database managed by a database server. For reasons similar to those discussed with respect to claim 18, the cited art fails to expressly or inherently describe all the features claim 18.

Based on the foregoing, Bridge fails to teach at least some features of claim 18, and therefore fails to teach all the features of claim 18. Therefore, claim 18 is patentable.

Rejection of Dependent Claim 20 Under 102(b)

Claim 20 recites wherein said "set of one or more files includes metadata describing database objects and commands for inserting data into the database objects, wherein the step of provisioning includes importing said database data into said database by executing said commands." Note that claim 18, upon which claim 20 depends, recites that the set of one or more files are caused to be transported between file systems by a database server. Files, that are transported between files systems by a database server, and that include commands executed to import database data, is a feature that is not disclosed or suggested in any way by the cited art.

The Office alleges that Bridge teaches this feature at col. 3, lines 25 – 34, reproduced below.

> Accordingly, there is a need for a way to increase the address space of disk pointers in an upward compatible manner. There is also a need for a way to transfer disk pointers between databases without patching.
>
> SUMMARY OF THE INVENTION
>
> In accordance with an aspect of the invention, a method of retrieving a data item from a computer database includes partitioning the database into a set of tablespaces and storing references to data items as tablespace-relative pointers, indicating a location relative to the tablespace containing the data item.

The passage above describes modifying disk pointers, transferring disk pointers, and partitioning data into tablespaces. It does not follow from these teachings that files are transported by a database server between file systems, and that the files include commands executed to import database data, as claimed.

Based on the foregoing, Bridge fails to teach at least some features of claim 20, and therefore fails to teach all the features of claim 20. Therefore, claim 20 is patentable.

Rejection of Dependent Claim 21 Under 102(b)

Claim 21 recites "said set of one or more files includes backup files created by a recovery manager, wherein the step of provisioning includes causing said recovery manager to create said database managed by said database server from said backup files." This limitation is not expressly or inherently described by the cited art.

The Office Action alleges that Bridge describes this feature at col. 6, line 64 to col. 7, line 5, reproduced below.

> When a disk pointer is read from a datafile, the database system can obtain a TSN for the disk pointer from the operating context. Therefore, the TSN of a tablespace-space DBA need not be stored in a disk pointer embedded in a datafile, allowing the portion of a disk pointer allocated for the AFN to indicate the TRFN instead. Such a disk pointer is a "tablespace-relative disk pointers." Disk pointers not embedded within a datafile, such as those disk pointers found in recovery logs, are stored with the proper TSN.

The above passage teaches that disk pointers in recovery logs are stored with the proper TSN (tablespace number). This fails to expressly or inherently describe creating a database from backup files. Those skilled in the art know that recovery logs are not backup files. Moreover, the passage does not in any way suggest much less disclose creating a database from recovery logs or backup files.

Based on the foregoing, Bridge fails to teach at least some features of claim 21, and therefore fails to teach all the features of claim 21. Therefore, claim 21 is patentable.

Rejection of Dependent Claim 22

Claim 22 recites "wherein an archive log stores data recording changes to said database made after creating the backup files, wherein the step of provisioning further includes changing said database to reflect changes recorded in said archive log." This feature is not suggested in any way much less disclosed by the cited art.

The teaching relied upon for rejecting claim 22 is the same as for claim 21, as described above. That passage does not even describe making changes to a database to reflect changes recorded in an archive log, much less where such a step is performed as part of provisioning database data included in one or more transported files, as claimed. (See claim 18, which claim 22 depends on).

Based on the foregoing, Bridge fails to teach at least some features of claim 22, and therefore fails to teach all the features of claim 22. Therefore, claim 22 is patentable.

Rejection of Dependent Claim 7 Based on 103

Claim 7 recites that the "database server provisions a synchronization mechanism that applies changes made to the tablespace to the copy."

"Section 103 forbids issuance of a patent when 'the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.'" KSR Int'l Co. v. Teleflex Inc., 127 S.Ct. 1727, 1734, 82 USPQ2d 1385, 1391 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art.... Graham v. John Deere Co., 383 U.S. 1, 17-18, 148 USPQ 459, 467 (1966). See also KSR, 127 S.Ct. at 1734, 82 USPQ2d at 1391. "If a

court, or patent examiner, conducts this analysis and concludes the claimed subject matter was obvious, the claim is invalid under §103."

In the present matter, the Examiner has made clearly erroneous factual findings regarding the scope and content of the prior art, and in particular, what certain cited prior art references teach. Here, the Examiner is alleging the Wang teaches that a "database server provisions a synchronization mechanism that applies changes made to the tablespace to the copy." However, this allegation is incorrect. Wang fails to even teach of a synchronization mechanism of claim 1, much less that a database server provisions such a synchronization mechanism, as claimed. Therefore, the Examiner's finding regarding the scope and content is erroneous, and the rejection based thereon is invalid.

The Examiner's error stems, at least in part, from a faulty interpretation, one that is based on the wrong sense of the term synchronization mechanism. As mentioned before, terms in computer technology have multiple meanings or senses, like many words of the English language. Synchronization mechanism is such a term. The sense of Wang is different than that explicitly and expressly required by claim 7.

In Wang, the term's sense is a synchronization mechanism that controls concurrent access to resources such as data. The synchronization mechanism described in the passage is a distributed lock manager that issues locks to synchronize access.

In claim 7, the term's sense is a synchronization mechanism that keeps bodies of data in sync. For example, "synchronization mechanisms can also be automatically provisioned to keep the tablespace and a copy in sync." (Application 0025) A defining feature of such a mechanism is that it "applies changes made to the tablespace to the copy", as claimed. Wang does not teach the kind of synchronization mechanism that has this express defining feature.

The Office Action alleges that the synchronization mechanism is not defined in a way that removes the reference from reading upon claims. However, this is incorrect. The synchronization mechanism, as expressly defined by claim 7, does not read on Wang.

First, the synchronization mechanism as claimed applies changes made to the tablespace to the copy of the tablespace. The synchronization mechanism of Wang only controls access to data, but does not make changes to data.

More importantly, the synchronization mechanism as claimed is provisioned by a database server. Wang does not teach that the synchronization mechanism it teaches about is provisioned by a database server.

Based on the foregoing, Bridge fails to teach at least some features of claim 7, and therefore fails to teach all the features of claim 7. Therefore, claim 7 is patentable.

CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that all the rejections lack the requisite legal and factual basis.

If any fee is missing or insufficient, the Director is hereby authorized to charge any applicable fee to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: March 10, 2008

/MarcelKBingham#42327/

Marcel K. Bingham
Reg. No. 42,327

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Tel: (408) 414-1080x208
Fax: (408) 414-1076

VIII. CLAIMS APPENDIX

1. (Previously Presented) A method for automatically provisioning data in a distributed database system, the method comprising computer-implemented steps of:
a database server causing a tablespace to be transported from a first file system to a second file system; and
after transporting said tablespace to said second file system, said database server importing said tablespace into a local database managed by said database server.

2. (Original) The method of claim 1, wherein the step of a database server causing a tablespace to be transported and the step of said database server importing said tablespace are both performed in response to invocation of a routine.

3. (Original) The method of claim 1, wherein said routine is written in code that conforms to a database language and that may be executed by a database server.

4. (Original) The method of claim 1, wherein the step of importing includes attaching said tablespace to said local database.

5. (Original) The method of claim 1, wherein the tablespace is attached to another database before and during performance of the step of said database server causing a tablespace to be transported.

6. (Original) The method of claim 1, wherein the tablespace is offline before and during performance of the step of said database server causing a tablespace to be transported.

7. (Original) The method of claim 1, wherein:
the step of importing the tablespace includes attaching a copy of the tablespace, wherein the copy is different than said tablespace; and

said database server provisions a synchronization mechanism that applies changes made to the tablespace to the copy.

8. (Original) The method of claim 7, wherein the synchronization mechanism applies changes made to the copy to the tablespace.

9. (Original) The method of claim 7, wherein the steps further include:
the synchronization mechanism determining which changes to the tablespace to propagate to the copy based on the results of an evaluation of a set of rules by a rules engine; and
wherein the step of provisioning the synchronization mechanism includes configuring said set of rules.

18. (Previously Presented) A method for automatically instantiating database data in a distributed database system, the method comprising computer-implemented steps of:
a database server causing a set of one or more files to be transported from a first file system to a second file system;
wherein said set of one or more files store said database data; and
after transporting said set of one or more files to said second file system, said database server provisioning said database data as at least part of a database managed by said database server.

19. (Previously Presented) The method of claim 18, wherein the set of files is a tablespace, wherein the step of provisioning includes:
attaching said tablespace to said database managed by said database server.

20. (Previously Presented) The method of claim 18, wherein said set of one or more files includes metadata describing database objects and commands for inserting data into the database objects, wherein the step of provisioning includes importing said database data into said database by executing said commands.

21. (Previously Presented) The method of claim 18, wherein said set of one or more files includes backup files created by a recovery manager, wherein the step of provisioning includes causing said recovery manager to create said database managed by said database server from said backup files.

22. (Original) The method of claim 21, wherein an archive log stores data recording changes to said database made after creating the backup files, wherein the step of provisioning further includes changing said database to reflect changes recorded in said archive log.

23. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.

24. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 2.

25. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 3.

26. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 4.

27. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 5.

28. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.

29. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.

30. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.

31. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.

40. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 18.

41. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 19.

42. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 20.

43. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 21.

44. (Previously Presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 22.

IX. EVIDENCE APENDIX

None.

X. RELATED PROCEEDINGS APENDIX

None.